

Compiler Optimization Levels on the ERDC Cray XT3

Tyler Simon

U.S. Army Engineer Research & Development Center
Major Shared Resource Center

Overview

- ERDC XT3 Intro
- Brief Description of Codes
- Compiler Options
- Results
- Conclusions

ERDC XT3 (Sapphire)

- 4128 64 bit 2.6Ghz Opterons (single core)
- XT/OS 1.4.43
- 2GB memory per processor
- Catamount on compute nodes
- March: Upgrade to dual core, OS 1.5.34a
- December: new XT4
- On to the codes!

MILC

- Quantum chromodynamics code
- Version 7
- 100% C
- Tests Improved action for Kogut Susskind quarks, RHMC algorithm
- Using 4.3GB lattice file as input

HYCOM

- Ocean modeling
- Version 2.1.25
- 31,000 Lines of code
- Fortran 90, MPI
- Test Case
 - 26-layer 1/4 degree global model
 - Simulates 1 day
 - Requires about 0.75 GB of memory per processor and about 4 GB of globally accessible scratch disk

GAMESS

- Quantum chemistry code
- Version: Sep 6 2006 (R4)
- 567,170 lines of code, 99% Fortran 77, 1% C
- Only library needed is tuned level 3 BLAS

GAMESS Test Case

- Performs a DFT to compute the nuclear gradient vector of a molecule
- Restricted Hartree-Fock calculation with self-consistent field wave functions
- Computes many energy integrals
- Molecular data in form of atom positions and electron orbitals

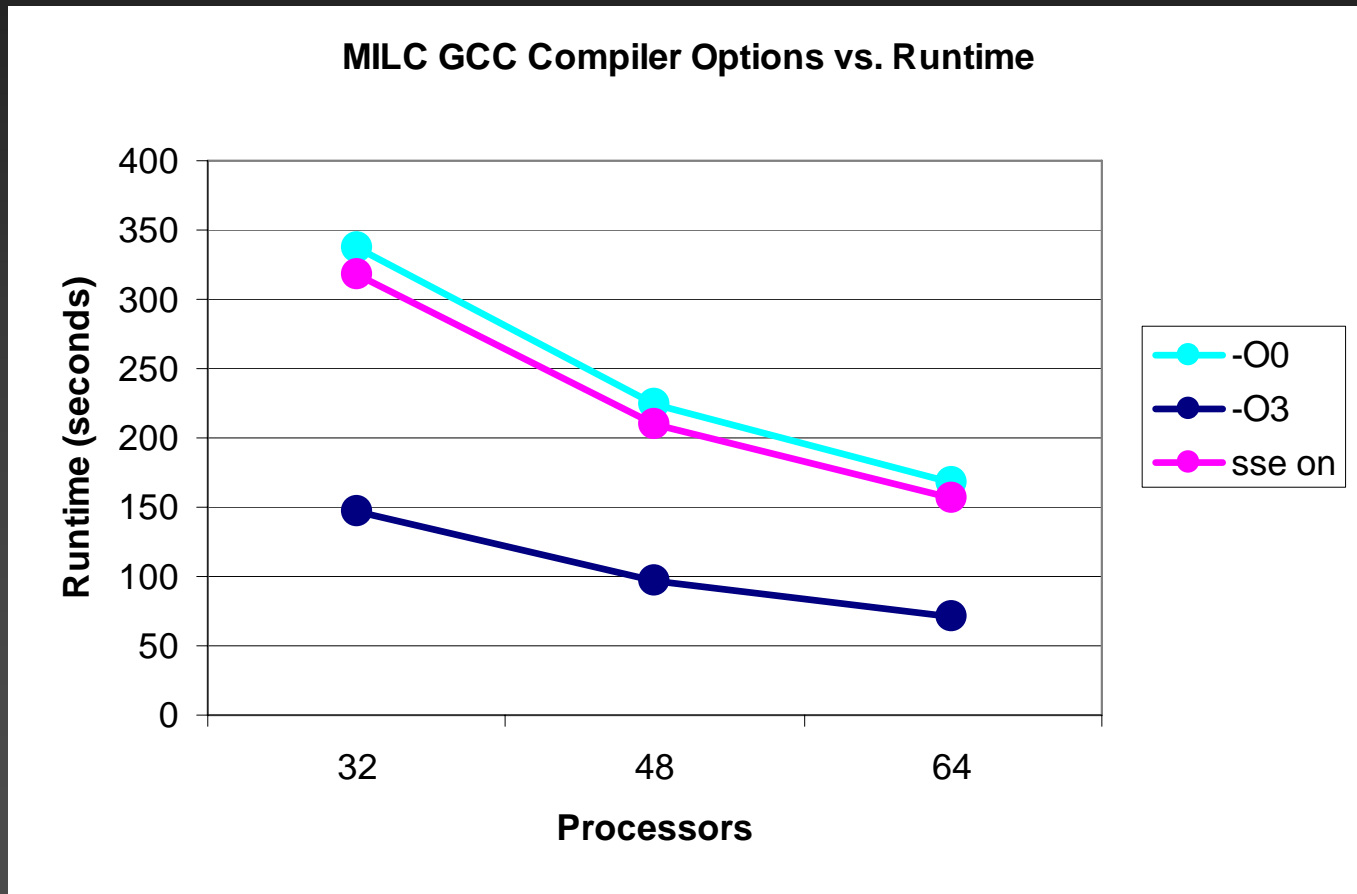
General Compiler Options

- PGI (6.1-4)
 - Fastsse
 - O0
 - O2 -Munroll=c:1 -Mnoframe -Mlre
 - O3
- GNU (3.2.3)
 - mfpmath=sse
 - O3
 - O0

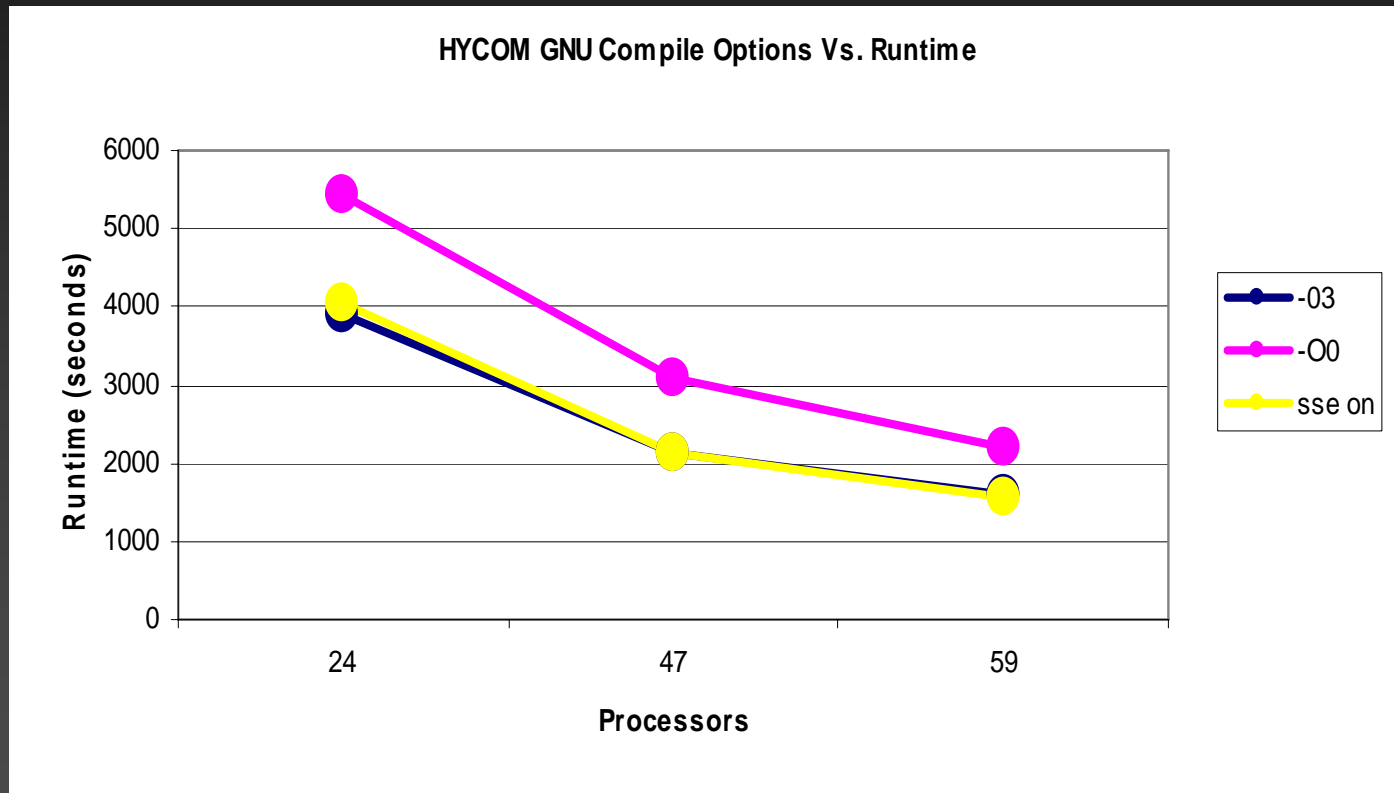
Compiling MILC

- `MPICH_CC=qk-gcc.`
 - No threading (pthreads or OpenMP).
 - No TCP/IP facilities (pipes, sockets or IP messages).
 - No `system()` calls.
 - No shared libraries, no `shmem`
 - No access to the `/proc` filesystem

MILC with GNU



HYCOM with GNU

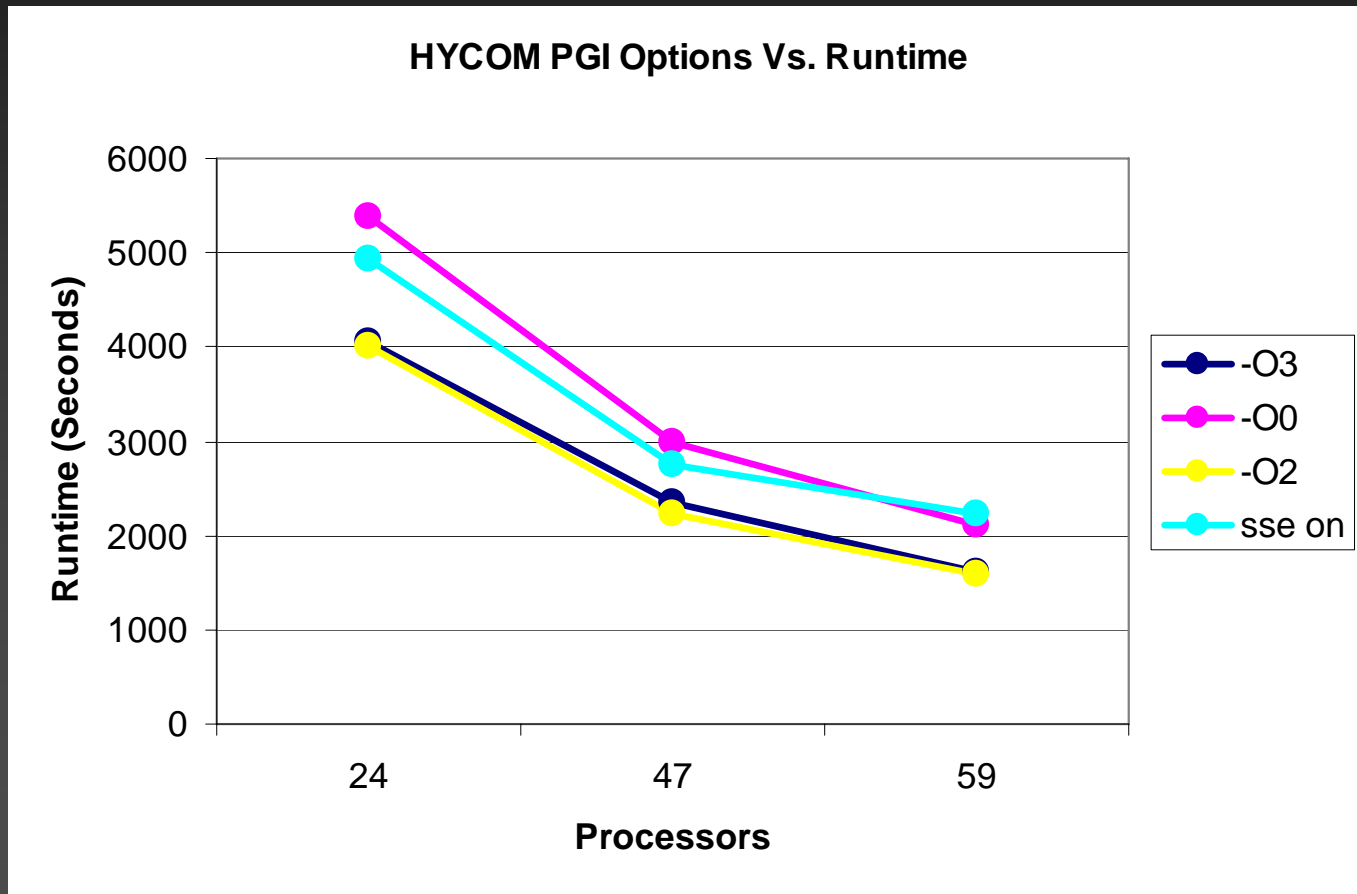


Compiling HYCOM with PGI

-O2 -Munroll=c:1 -Mnoframe -Mlre (non-sse)

- -Munroll=c completely unroll loops with this loop count or less
- For -Munroll=n:(m) says unroll other loops m times
- -Mnoframe does not set up a stack frame
- -Mlre is loop-carried redundancy elimination

HYCOM with PGI



Compiling GAMESS

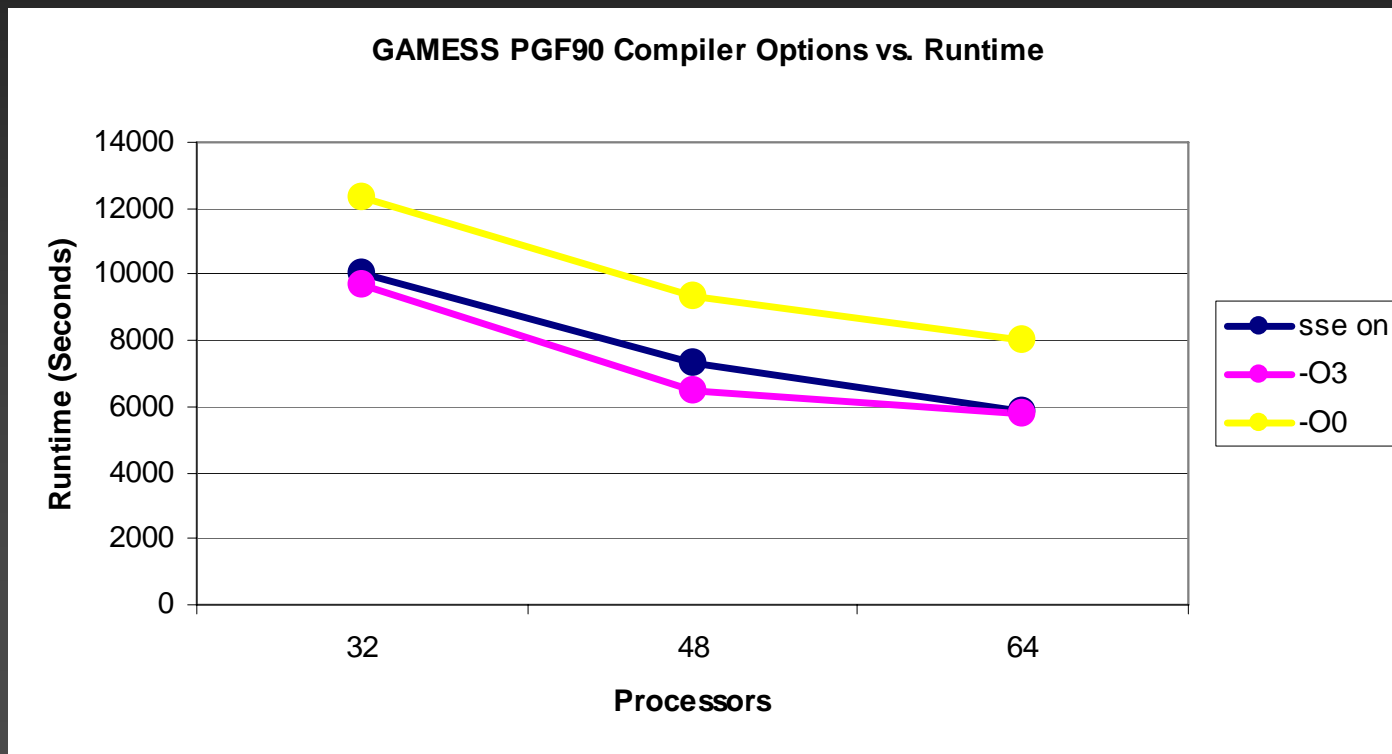
(explicit sse-on) fastsse –Munroll

-O0

-O3

- Distributed data interface front-end compiled with f77
- Pure-mpi

GAMESS with PGI



Conclusions

- HYCOM
 - may get a performance improvement with SSE for large (256p+) jobs
- MILC
 - Little improvement using Gnu SSE, recommend -O3
- GAMESS
 - No drastic improvement in using SSE